COMP90042

Web search and text analysis

Workshop Week 4

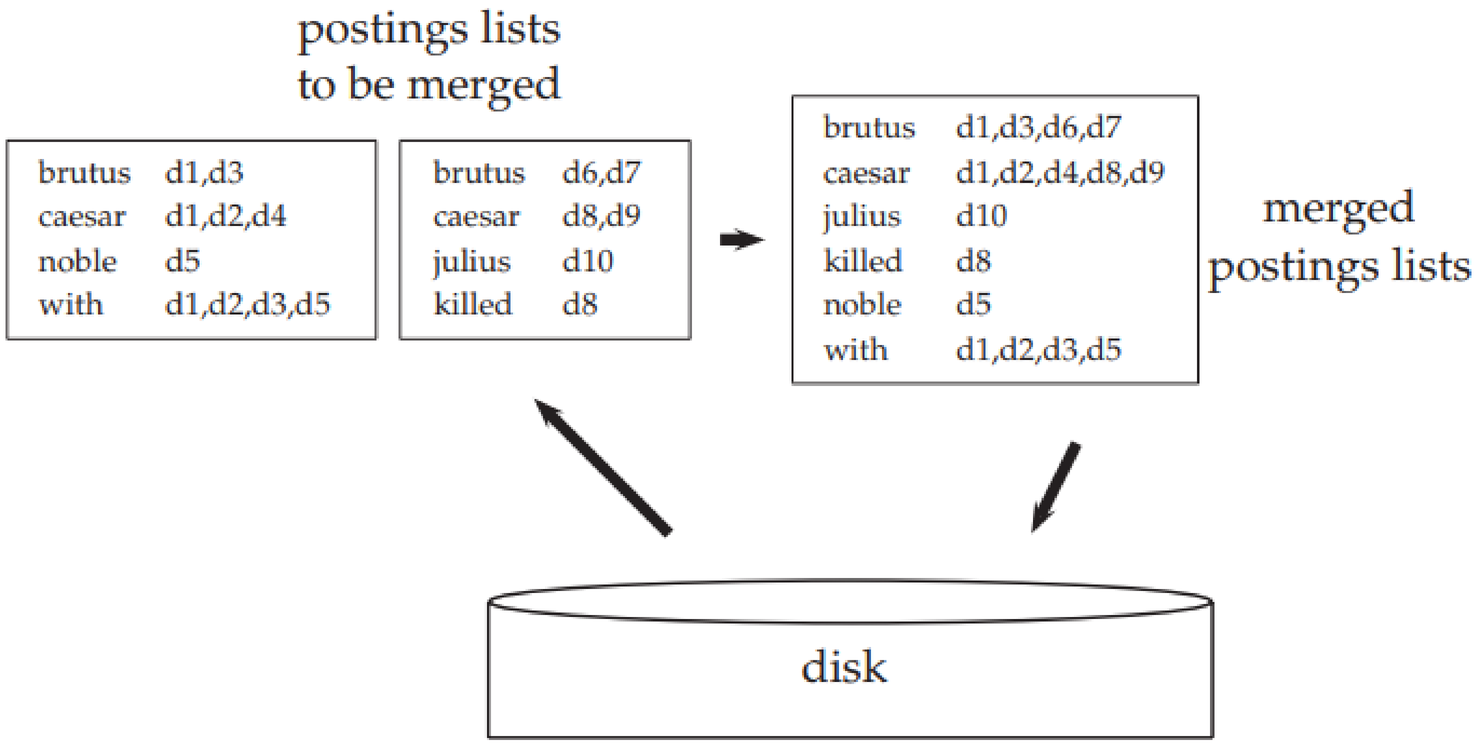
Your tutor

– Winn Chow (Senior Tutor)
– winn.chow1@unimelb.edu.au
– Office: Doug McDonell - 9.23
– Here, you can find my workshop slides:
– https://github.com/winnchow/COMP90042-Workshops

Q1

1. Discuss the process of static **inverted index construction** and how it can be used to perform in incremental index construction.

Static Inverted Index Construction



► **Figure 4.3** Merging in blocked sort-based indexing. Two blocks ("postings lists to be merged") are loaded from disk into memory, merged in memory ("merged postings lists") and written back to disk. We show terms instead of termIDs for better readability.

Dynamic Inverted Index Construction

- a large main index (M postings)
- a small auxiliary index (n postings)
- Searches are run across both indexes and results merged
- A merge requires merging and writing M + n postings to disk (I/O)

Q2

2. Why is a **logarithmic** index layout useful? What are the disadvantages of such an index structure?

Logarithmic index layout

- If we keep a logarithmic index layout, then
- The M postings on disk are in multiple indexes with spaces of 2n, 4n, 8n, 16n … postings
- Less I/Os are required to merge
- Disadvantage: Have to search multiple indexes and merge the search result

Q3

3. Based on the following top-6 retrieval results from a collection of 100 documents, and the accompanying binary relevance judgements

doc	score	relevance
a	0.4	0
b	1.2	0
c	2.2	1
d	0.5	1
e	0.1	1
f	0.8	0

compute the following evaluation metrics:

(a) precision@3

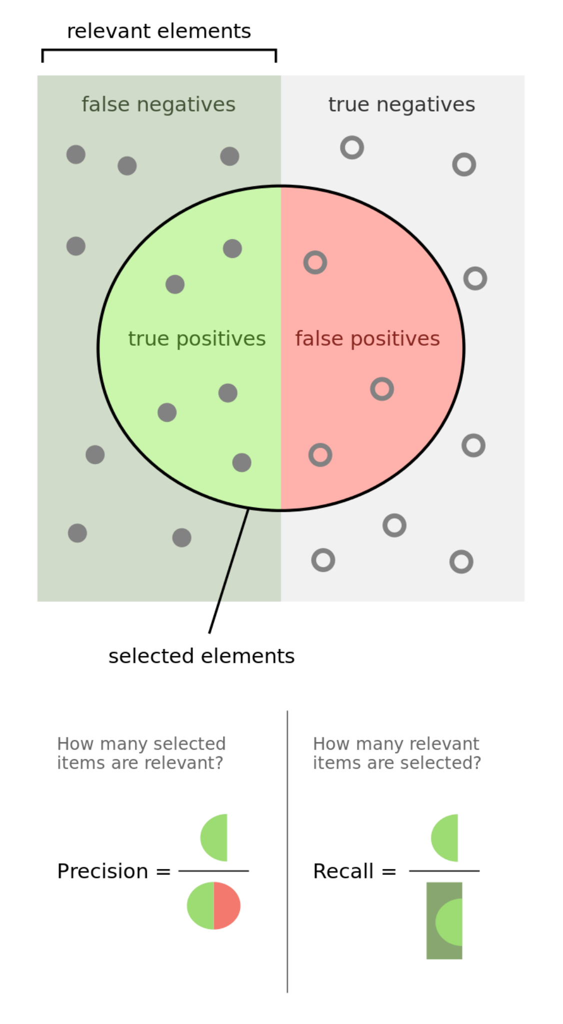
(b) average precision (do you need to make any assumptions about the document collection?); and

(c) rank-biased precision (RBP), with $p = 0.5$

(d) plot the precision-recall graph, where you plot (precision, recall) point for the top k documents, $k = 1, 2, \ldots 6$.

(e) what are the strengths and weaknesses of the methods above for evaluating IR systems?

3(a) precision@3



– Step 1: Rank by document scores

doc	score	relevance
a	0.4	0
b	1.2	0
c	2.2	1
d	0.5	1
e	0.1	1
f	0.8	0

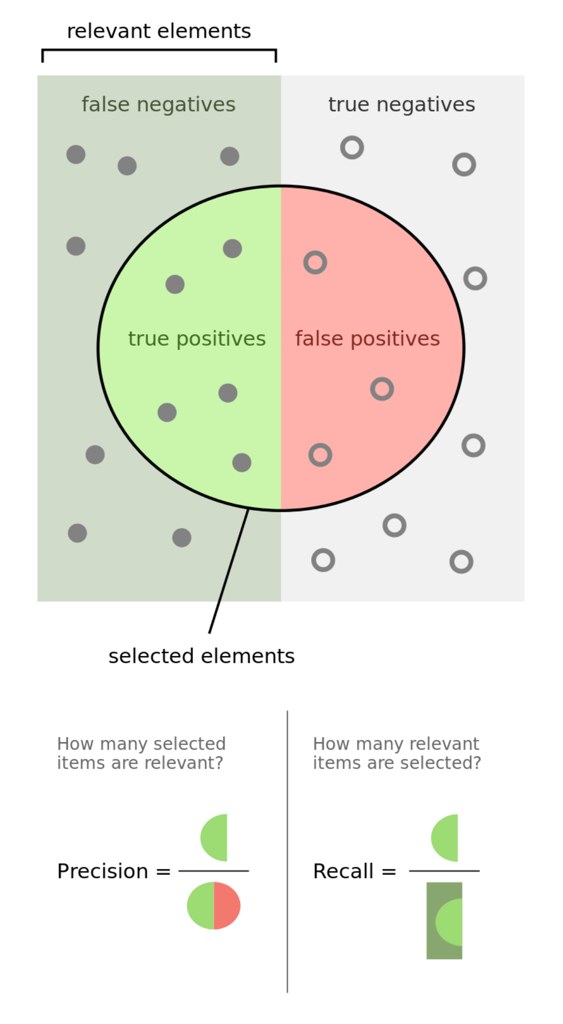
Rank (k)	Precision@k	Recall@k
5		
2		
1		
4		
6		
3		

– precision@k = $\frac{\sum_{i=1}^{k} relevance_i}{k}$, recall@k = $\frac{\sum_{i=1}^{k} relevance_i}{Total\ number\ of\ relevant\ documents}$

– precision@3 = precision using only documents ranked 1, 2, 3

– recall@3 = recall if we only return documents ranked 1, 2, 3

3(a) precision@3



– Step 1: Rank by document scores

doc	score	relevance
a	0.4	0
b	1.2	0
c	2.2	1
d	0.5	1
e	0.1	1
f	0.8	0

Rank (k)	Precision@k	Recall@k
5	2/5	2 / 3
2	½	1 / 3
1	1/1 = 1	1 / 3
4	2/4 = ½	2 / 3
6	3/6 = ½	3 / 3 = 1
3	1/3	1 / 3

– precision@k = $\frac{\sum_{i=1}^{k} relevance_i}{k}$, recall@k = $\frac{\sum_{i=1}^{k} relevance_i}{Total\ number\ of\ relevant\ documents}$

– precision@3 = precision using only documents ranked 1, 2, 3

– recall@3 = recall if we only return documents ranked 1, 2, 3

3(b) average precision

– average precision over relevant documents

$$= \frac{\sum_k precision@k \times relevance_k}{Total\ number\ of\ relevant\ documents}$$

3(b) average precision

– average precision = (precision@1 + precision@4 + precision@6) / 3

$$AP = \frac{1}{3} \times (\frac{1}{1} + \frac{2}{4} + \frac{3}{6}) = \frac{1}{3} \times 2 = \frac{2}{3}$$

3(c) rank-biased precision (RBP)

- RBP Formula (r_i is the ith element of the relevance vector of length d)

$$RBP = (1 - p) \times \sum_{i=1}^{d} r_i \times p^{i-1}$$

  - *p* is the persistence probability
  - *p* = 0.5

doc	score	relevance
a	0.4	0
b	1.2	0
c	2.2	1
d	0.5	1
e	0.1	1
f	0.8	0

Rank (i)	P^{i-1}
5	
2	
1	
4	
6	
3	

3(c) rank-biased precision (RBP)

- RBP Formula (r_i is the ith element of the relevance vector of length d)

$$RBP = (1-p) \times \sum_{i=1}^{d} r_i \times p^{i-1}$$

  - *p* is the persistence probability
  - *p* = 0.5

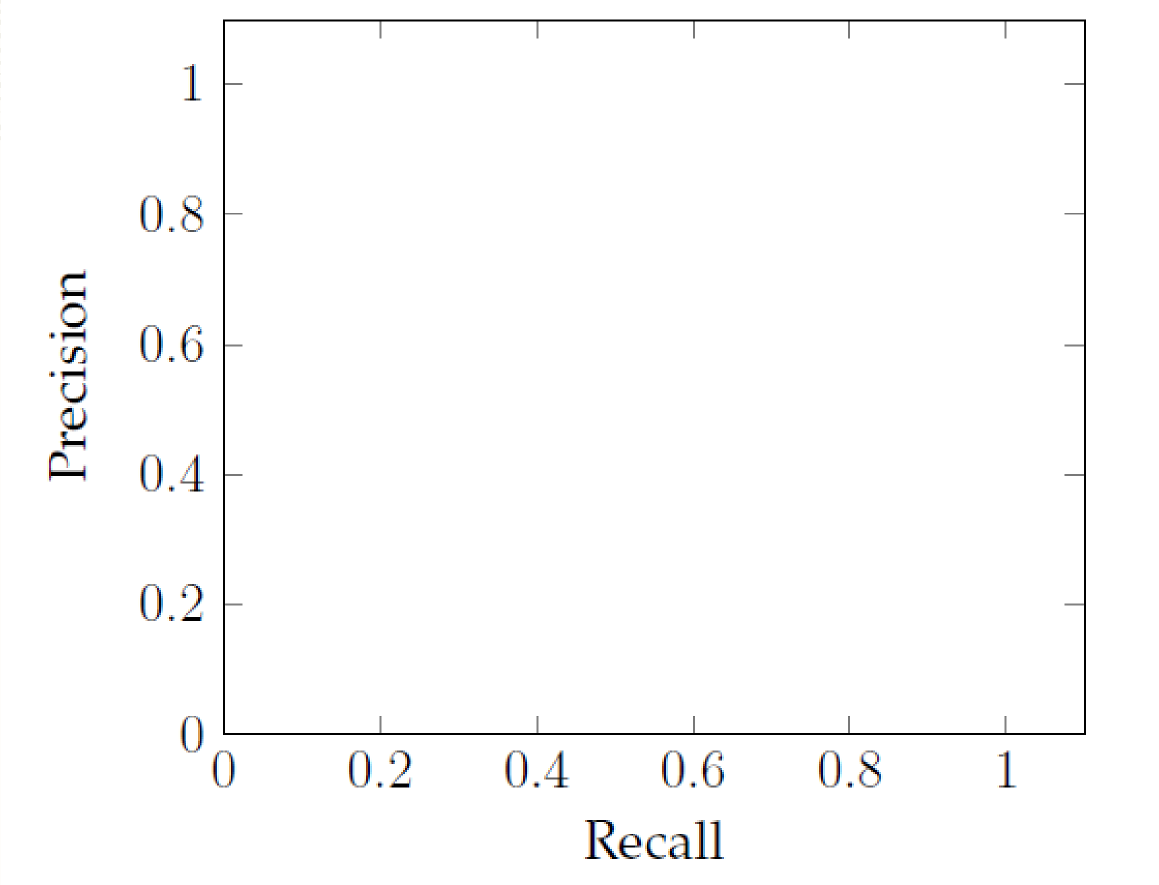
doc	score	relevance
a	0.4	0
b	1.2	0
c	2.2	1
d	0.5	1
e	0.1	1
f	0.8	0

Rank (i)	p^{i-1}
5	0.5^4
2	0.5^1
1	$0.5^0 = 1$
4	0.5^3
6	0.5^5
3	0.5^2

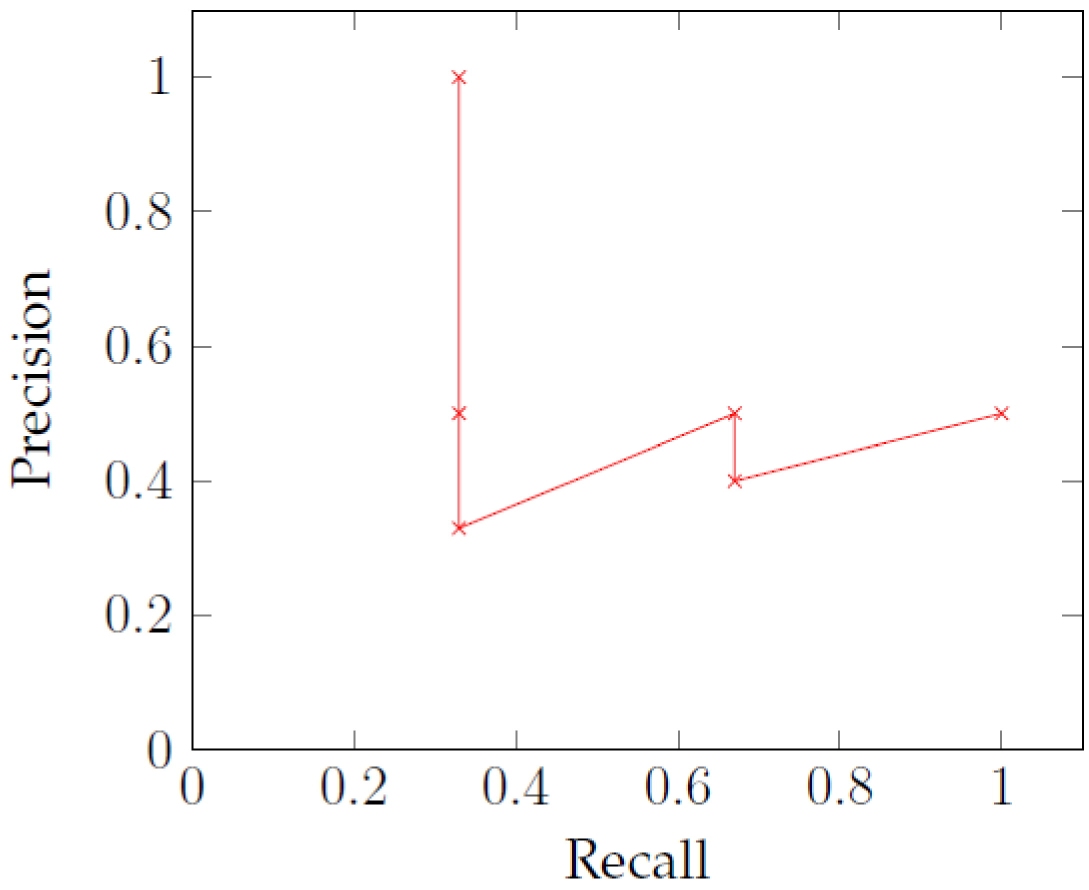
3(c) rank-biased precision (RBP)

$$
\begin{aligned}
RBP &= (1-p) \times \sum_i r_i p^{i-1} \\
&= (1-0.5) \times \left(0.5^0 + 0.5^3 + 0.5^5\right) \\
&= \frac{1}{2} \times \left(1 + \frac{1}{8} + \frac{1}{32}\right) \\
&= \frac{1}{2} \times \frac{1}{32} \times (32 + 4 + 1) \\
&= \frac{37}{64}
\end{aligned}
$$

3(d) plot the precision-recall graph



3(d) plot the precision-recall graph



3(e) what are the strengths and weaknesses of the methods?

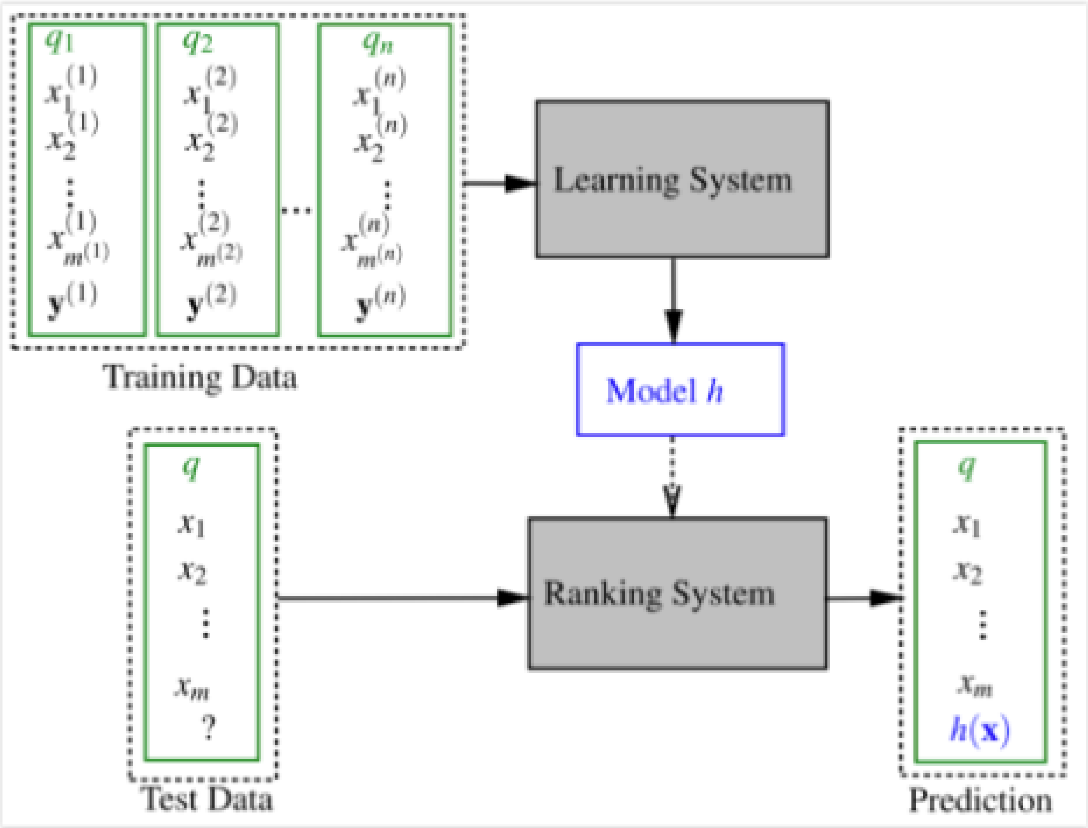
- precision@k = $\frac{\sum_{i=1}^{k} relevance_i}{k}$
  - Easy to evaluate and understand
  - But no differentiation by rank for ranked document 1, 2, …, k
  - But no adjustment for the size of the relevant documents
- average precision = $\frac{\sum_k precision@k \times relevance_k}{Total\ number\ of\ relevant\ documents}$
  - Differentiation by rank
  - Adjustment for the size of the relevant documents
  - But need to know the size of the relevant documents
- Rank biased precision $RBP = (1-p) \times \sum_i r_i p^{i-1}$
  - Differentiation by rank
  - Adjustment for the size of the relevant documents
  - But need to decide on the persistence probability *p*

Q4

4. How can a retrieval method be learned using supervised machine learning methods? Consider how to frame the learning problem, what data will be required for supervision, and what features are likely to be useful.

Q4 Learning to Rank

- *n queries*
- *m* documents
- *y* is the relevance judgement



Taken from: Tie-Yan Liu: Learning to Rank for Information Retrieval

Q4 Learning to Rank

- User features e.g. search history of the user, location
- Document features e.g. page rank, quality score, topics
- Query features e.g. number of query terms, popularity of the query